

# Segmentation-based Multi-Scale Edge Extraction to Measure the Persistence of Features in Unorganized Point Clouds

Dena Bazazian<sup>1</sup>, Josep R. Casas<sup>2</sup>, Javier Ruiz-Hidalgo<sup>3</sup>

*Signal Theory and Communications Department, Universitat Politècnica de Catalunya, Barcelona, Spain*

<sup>1</sup>dena.bazazian@tsc.upc.edu, <sup>2</sup>josep.ramon.casas, <sup>3</sup>j.ruiz}@upc.edu

**Keywords:** Edge extraction, Multi-scale, Segmentation, Unorganized point cloud

**Abstract:** Edge extraction has attracted a lot of attention in computer vision. The accuracy of extracting edges in point clouds can be a significant asset for a variety of engineering scenarios. To address these issues, we propose a segmentation-based multi-scale edge extraction technique. In this approach, different regions of a point cloud are segmented by a global analysis according to the geodesic distance. Afterwards, a multi-scale operator is defined according to local neighborhoods. Thereupon, by applying this operator at multiple scales of the point cloud, the persistence of features is determined. We illustrate the proposed method by computing a feature weight that measures the likelihood of a point to be an edge, then detects the edge points based on that value at both global and local scales. Moreover, we evaluate quantitatively and qualitatively our method. Experimental results show that the proposed approach achieves a superior accuracy. Furthermore, we demonstrate the robustness of our approach in noisier real-world datasets.

## 1 INTRODUCTION

The computer vision community has drawn attention to 3D scene analysis in recent years, with data captured with stereo and multi-camera systems, and especially after the success of commercial depth sensors, such as MS Kinect or Asus Xtion. The data acquired by these devices, however, may contain noise, outliers and, even if the projected points are uniformly distributed on the image (at the pixel level), the distribution tends to be highly non-uniform in the 3D point cloud. The situation becomes more complicated since the point clouds are unorganized: the points in the cloud are not initially connected, and information about surface normals has to be computed with a certain amount of uncertainty. For these reasons, the shape analysis of the measured data-set becomes highly challenging in computer graphics and computer vision.

Our research deals with the edge extraction problem in surfaces represented by point clouds. Estimating and identifying edges enables a better understanding of the structural features of the underlying surfaces. An edge is formed when two surfaces, which we may approximate by planes, with sufficiently different orientation meet. Hence, the planes defined by normal vectors of the surrounding points should be estimated in order to determine an edge point. A

region of contiguous edge points is then defined as an edge region in the surface represented by the unorganized cloud.

Edge extraction identifies crucial characteristics of the underlying geometry. It can be used to detect feature points, and improves the performance of geometry processing such as adaptive sampling, segmentation, detection and scene analysis.

For edge extraction techniques, multi-scale schemes improve noise sensitivity and add robustness against the scale dependency of features. The rationale behind this is to get an optimal scale for each point in order to analyze the intrinsic structure around it.

In our approach, we segment the surface to explore global classification of edges and feature points. One of the aspects of the multi-scale analysis is to cope with false feature measurements arising from the fact that part of the surface intersects the local neighborhood.

The main contributions of this paper are the following. First, we put forward a new classification framework that allows for discrete surface analysis at multiple scales. Second, we propose an optimal edge extraction technique by considering a segmentation of the surface in order to obtain a multi-scale description of edges in the point cloud. Third, we improve the accuracy of the edge extraction technique in

unorganized point clouds.

The remainder of the article is organized as follows. Section 2 presents related work, followed by a description of our approach and architecture in Section 3. Section 4 reports the experimental results of our approach, and conclusions are drawn in section 5.

## 2 RELATED WORK

The state of the art for edge extraction in point clouds is summarized below. We group the various contributions according to the different aspects of the problem: extraction of sharp features, estimation of normals, segmentation and multi-scale approaches.

### 2.1 Sharp feature extraction

Robust statistics have been exploited to extract sharp features (Fleischman et al., 2005; Daniels et al., 2008; Oztireli et al., 2009). A technique for edge detection for the registration of point clouds was introduced by Choi (Choi et al., 2013). Other authors have explored just the detection of edge points (Sidiropoulos and Lakakis, 2016) or the segmentation of larger geometric structures such as surfaces (Demarsin et al., 2007; Xu et al., 2015), lines (Lin et al., 2015) and, more recently, contour detectors based on classifiers (Hackel et al., 2016) have also been proposed.

In unorganized point clouds, the geometry defined by the neighborhood of a point is almost the only available information to extract features from. In this line, several authors (Weber et al., 2010; Weber et al., 2012; Gumhold et al., 2001; Feng et al., 2014) propose a region growing method that decomposes the point cloud into clusters, and identifies the regions with sharp features based on the analysis of the normals. In these approaches, extracting sharp edge features from a 3D point cloud requires the computation of accurate normals from the neighborhood to generate high quality surfaces. This brings up the problem of normal estimation, which is closely related to sharp feature extraction, as we will discuss next.

### 2.2 Normal estimation

Regression based estimation was first proposed by Hoppe (Hoppe et al., 1992). For each point in the cloud, a least squares local plane is fitted to its  $k$  nearest neighbors using PCA. The normal at each point is the eigenvector corresponding to the smallest eigenvalue of the covariance matrix (Lange and K., 2005). By assigning Gaussian weights to the neighbors of

each point, Pauly and Gross (Pauly et al., 2003b; Gross and Pfister, 2007) proposed a weighted version of this basic approach. For noisy point clouds, Mitra (Mitra and Nguyen, 2003) suggested an adaptive neighborhood size based on local properties, such as noise scale, curvature and sampling density. A similar motivation led Wang (Wang and Feng, 2015) to propose an algorithm to develop a normal estimation method that rejects neighborhood outliers.

### 2.3 Segmentation

Various algorithms have been proposed for point cloud segmentation as an indirect approach to extract edges. Some methods are based on the surface such as those by Zhang and Reisner (Zhang et al., 2013; Reisner-Kollmann and Maierhofer, 2012), which attempt to find sets of points in the scene that fit planes and primitives. Moreover, Rabbani and Son (Rabbani et al., 2006; Son and Kim, 2013) proposed a graph based method which do not constrain users to modeling with primitives. In addition, Liu (Liu and Youlun, 2008) proposed a mapping of the normal of a point cloud into a Gaussian sphere, producing a Gaussian image; afterwards this spherical image can be clustered to identify shapes. Furthermore, Kustra (Kustra et al., 2014) proposed to find the most probable local quasi-flat surface patches passing through each point by using a clustering approach, and then merging these patches to classify the input points into manifolds or noise.

### 2.4 Multi-scale

Defining a proper neighborhood for each point in the cloud raises the question of finding the right scale factor. Scale in computer vision has always been of extreme importance, more so with true space 3D data as in point clouds. It constitutes a limiting factor in the automatic estimation of a point feature representation. Pauly (Pauly et al., 2003a) proposes a surface variation method based on PCA at multiple scales, and evaluates the likelihood of a point belonging to a feature. Instead of surface variation, Ho (Ho and Gibbines, 2009) adopts the rotation and translation invariant local surface measure as a local geometric property to compute the feature confidence value and suitable scales. In addition, according to the significance of the normals in the point cloud, the authors in (Ioannou et al., 2015) proposed a multi-scale approach by computing the difference of normals. Furthermore, to overcome the complexity of the noise in the point cloud, Park (Park et al., 2012) proposes another multi-scale technique based on tensor voting for

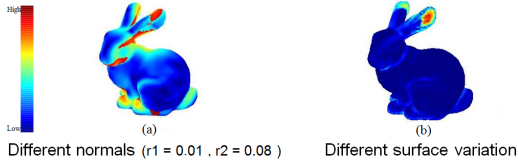


Figure 1: (a) The difference of normals (Ioannou et al., 2015), and (b) the difference of surface variation (Pauly et al., 2003a), illustrated with a jet-color map

point clouds that contain random noise.

## 2.5 Our contribution

Our goal is to analyze the persistence of a selected unique segment by using different distances over multiple scales. The method we propose is designed to extract edges in unorganized point clouds reliably and accurately. In our work, we consider the segmentation method of Kustra (Kustra et al., 2014) in order to explore a global classification. Our contribution is a new classification framework by segmentation which allows for discrete surface analysis at multiple scales with the purpose of improving the accuracy of edge extraction. The advantage of this approach is to cope with false feature measurements when another surface intersects the local neighborhood.

## 3 PROPOSED APPROACH

The following two subsections describe our proposal. We present the motivation and the general idea next, and then discuss the main technique for the segmentation-based multi-scale edge extraction, which is depicted in Algorithm 1.

### 3.1 Motivation and general idea

We define a multi-scale operator based on the difference of normals according to Ioannou (Ioannou et al., 2015) by using the estimated surface normal map of a point cloud. Figure 1 (a) shows the difference of normals when estimated for two different neighborhood sizes. In order to measure the persistence of a feature over all scales, we compute the surface variation (Pauly et al., 2003a) for each point at different scales, as shown in Figure 1 (b).

If the structure of the larger neighborhood around a center point is significantly different from that of a smaller neighborhood, then the direction of the two estimated normals is likely to vary by a large amount. In that case, a value between the two radii is often a representative of the scale near the center point. We

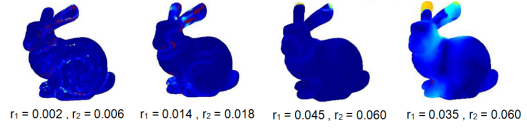


Figure 2: The difference of normals computed for a series of radii couples, illustrated with a jet-color map

illustrate how the difference of normals behaves for a variety of neighborhood sizes in Figure 2.

When increasing the values of the scale parameter, the computed difference of normals may be unreliable in case that a foreign surface intersects the local neighborhood. In order to cope with this issue, we consider Kustra’s global classification technique (Kustra et al., 2014). As shown in Figure 3, it is possible to segment the surface in order to obtain a global analysis of the point cloud.

When the local analysis neighborhood in the point cloud is defined by Euclidean distance, part of a foreign surface may be considered in the local neighborhood. For instance, for a sample point located on the bunny’s ear, some neighbors at large scales may come from other surfaces, as shown in Figure 4 (a). In order to overcome this drawback, we propose to replace the Euclidean by geodesic distance, as shown in Figure 4 (b). This approach also allows the global classification of the whole point cloud considering the different patches of the segmentation, as shown in Figure 4 (c).

### 3.2 Overview of our algorithm

Our main objective is to face the challenge in edge extraction related to varying neighborhood sizes for multi-scale analysis and, in particular, to avoid considering foreign surface patches in the local neighborhoods. We consider three steps in the proposed strategy: first, we segment the point cloud into different patches in order to avoid considering non-local points in the neighborhood; second, we propose a multi-scale operator to explore the scale of each segment; third, we extract edges according to the appropriate neighbor size of each segment. The details of the proposed technique are explained next and presented in Algorithm 1.

#### 3.2.1 Segmentation

The first step in our algorithm is the global segmentation of the point cloud. We use the technique proposed in (Kustra et al., 2014), which considers the global connectivity structure of the point cloud. Ac-

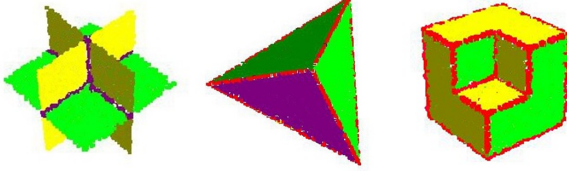


Figure 3: Extracting manifolds from the point cloud (Kustra et al., 2014)

cording to Kustra’s method, we first cluster the Gauss map of the neighborhood by using the geodesic distance on the Gaussian sphere between the normals of the map. Then, we extract the curved manifolds from the patch connectivity graph via a multiple-source flood fill. Segmentation results look like shown in Figs. 3 and 4 (c).

### 3.2.2 Multi-scale

In order to find out the adaptive neighborhood size on the point cloud we consider a multi-scale method according to the size of each segment. First, the area of each segment in the point cloud is calculated according to the position of the associated points in the 3D Cartesian coordinate system. Afterwards, we explore the adaptive scale by considering an iterative procedure for each segment. We start from a small scale such as 0.1 times the area (this value was a proportionate scale in our experiments with various shapes). Then, we compute the probability of being an edge for each sample point according to the surface variation method (Pauly et al., 2003a), which is based on the eigenvalues of the covariance matrix. The surface variation  $\sigma_n(p_i)$  at point  $p_i$  with neighbor size of  $n$  is defined as:

$$\sigma_n(p_i) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \quad (1)$$

where  $\lambda_0$ ,  $\lambda_1$  and  $\lambda_2$  are the eigenvalues of the covariance matrix of  $p_i$  and  $\lambda_0 \leq \lambda_1 \leq \lambda_2$ .

The surface variation  $\sigma_n(p_i)$  for each sample point  $p_i$  allows to distinguish the points belonging to a flat surface from those belonging to an edge in the point cloud. Since the smallest eigenvalue of the covariance matrix for flat surfaces is zero, then the value of the surface variation for flat surfaces would also be zero. Accordingly, we define the probability of being an edge as:

$$P(p_i) = \frac{\sigma_n(p_i)}{\Theta_n(S)}, \quad (2)$$

where  $P(p_i)$  is the probability of being an edge for the sample point  $p_i$  in the considered segment, and  $\sigma_n(p_i)$

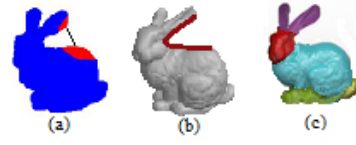


Figure 4: Estimating the neighbors of a sample point on the ear of *bunny* at large scales: (a) far away neighbors may belong to foreign surfaces when Euclidean distance is used; (b) geodesic distance is a better choice to explore large local neighborhoods; (c) the point cloud can be segmented to distinguish different surfaces.

is the surface variation of  $p_i$ . The value  $\Theta_n(S)$  in the denominator stands for the largest surface variation for the running segment of the point cloud. Thus, if the surface variation is zero, then that sample point is definitely not an edge. The larger the surface variation, the higher the probability of that sample point being an edge. This procedure is then repeated at progressively larger scales of the running segment, in order to measure the persistence of the  $p_i$ ’s feature according to the probability variation, as described in Algorithm 1- procedure 2.

### 3.2.3 Edge extraction

As a last step, we extract edges according to the fast technique proposed in (Bazazian et al., 2015), which is also robust for edges in very small dihedral angles. Bazazian’s edge extraction technique analyzes the variation of eigenvalues of the covariance matrix (via the surface variation parameter); afterwards, the points with zero or almost zero surface variation are considered as non-edge points. Instead of considering a single scale for whole the point cloud as in (Bazazian et al., 2015), we are able to improve the fast and robust edge extraction by defining an appropriate local scale for each segment of the point cloud via our multi-scale approach.

## 4 EXPERIMENTAL RESULTS

We performed experiments with synthetic point cloud geometries, as those in Figure 3 and 10, and classical point cloud models, as those in Figure 5. Synthetic geometries allow estimating the accuracy of the algorithm according to a known ground-truth. Figure 5 displays progressive iterations in the estimation of an adaptive scale for the three point cloud models of Bunny, Buddha and Dragon.

In order to explore the appropriate neighborhood size, we describe the application to the Bunny model. To this end, first we segment the point cloud into the

---

**Algorithm 1** Segmentation-based Multi-Scale Edge Extraction

---

```
1: procedure 1: SEGMENTATION
2: Input: A point cloud  $\mathcal{C}$  which samples a surface  $S \subset \mathbb{R}^3$ 
3: Output: A set of segment manifolds  $\mathcal{S}$  of  $\mathcal{C}$ 
4:   for all points  $v \in \mathcal{C}$  do
5:     Estimate the normals at  $v$ , call them  $n_v$ 
6:   end for
7:   Create a graph  $G = (V, E)$  with  $V = \mathcal{C}$  and initially  $E := \emptyset$ 
8:   for all  $v \in \mathcal{C}$  do
9:     for all  $w \in \mathcal{C}$  such that  $v$  and  $w$  are neighbors in  $\mathcal{S}$  do
10:      compute the angle between  $\angle n_v, n_w$ 
11:      if  $\angle n_v, n_w \leq T$  for some threshold value  $T$  then
12:         $E := E \cup \{v, w\}$ 
13:      end if
14:    end for
15:  end for
16:  Explore a flood-filling over the patch connectivity stored in  $G$  and compute segments  $\mathcal{S}$ 
17: end procedure 1
18: procedure 2: MULTISCALE
19: Input: A segment  $S \in \mathcal{S}$ 
20: Output: The adaptive scale (neighborhood size)  $N_S$ 
21:    $area_S := \text{area of } S, N_S := 0, p_S^0 := 0, \Delta := 1000$ 
22:   for  $i = 1$  to  $10$  do
23:     Let  $N_S^i := (0.1 * area_S) + (i * 2 * (area_S/100))$ 
24:      $p_S^i :=$  the probability of being an edge in  $\mathcal{C}$  at level  $N_S^i$  computed according to the surface variation
25:     Compute persistence:  $\Delta_i := p_S^i - p_S^{i-1}$ 
26:     if  $\Delta_i \leq \Delta$  then
27:        $\Delta := \Delta_i, N_S := N_S^i$ 
28:     end if
29:   end for
30:   Return  $N_S$ 
31: end procedure 2
32: procedure 3: EDGE EXTRACTION
33: Input: A segment  $S \in \mathcal{S}$  and  $N_S$ 
34: Output: set of edge points  $F$  of  $S$ 
35:    $F := \emptyset$ 
36:   for each sample point  $P \in S$  do
37:     Compute the surface variation  $V_P$  at  $P$  at scale  $N_S$ 
38:     if  $V_P \geq T'$  for some threshold value  $T'$  then
39:        $F := F \cup \{P\}$ 
40:     end if
41:   end for
42: end procedure 3
```

---

four patches. Next, according to the iterations which have been explained in section 3.2, we explore the best adaptive scale to extract the edges in the point cloud. The appropriate neighborhood size for each segment will depend on its area. Figure 6 illustrates the variation of neighborhood sizes of each Bunny's segment at each iteration. In this plot we show the average number of points which are included in each radius surrounded by a sample point. The plot demonstrates the sensitivity of determining the neighborhood size for small segments of a point cloud, which leads to considering almost all the points of the segment as neighborhood points.

Moreover, Figure 7 illustrates how the different Bunny segments behave differently for edge detection with varying neighborhood sizes. We can observe that when increasing the size of the neighborhood, the number of extracted edge points increases in all the segments. The figure shows also that at some neighborhood sizes, such as for instance 0.95, there is a larger probability of false positives for small segments than for large segments. This also proves the need of the multi-scale approach, as edge detection with the same neighborhood size for all the segments of a point cloud (which have different actual scales) is prone to error. Hence, we propose to define the neighborhood size for each segment proportional to its scale, in order to get more precise results of edge extraction.

For the comparison of the percentage of edge points, we have estimated the probability of being an edge depending on the various segments of Bunny across the various scales. The probability of being an edge is computed according to the fluctuations of the surface variation at each segment. As shown in Figure 11, for those segments with small area (ear and head) at large scales (neighbor size 1.35 or 1.50), edge points are either undetected or just very few are detected. Almost all points of small area segments are detected as edge points at large scales. In these plots we demonstrate the different behavior of the variation of neighborhood sizes according to the various segments of a single shape. We illustrate that considering an appropriate neighbor size for each segment of a point cloud is essential for edge detection.

For the quantitative evaluation of the proposed algorithm, we perform experiments with artificial geometric point clouds such as the intersection of three planes in Figure 3 (left). In this point cloud each plane is considered as a single segment of unit area. In Figure 8 we show ten iterations for exploring the adaptive scale for this surface, by computing the

F1-score defined as:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

where *Relevant* is defined as the points which are labeled as ground-truth, whereas *Precision* is defined as how many selected points are relevant and *Recall* is defined as how many relevant points are selected. Precision and Recall are computed as:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad (4)$$

where *TP* stands for True Positives representing the number of correctly detected points, *FP* stands for False Positives representing the number of wrongly detected points, *FN* stands for False Negatives, representing the number of false rejections, i.e. points that belong to the ground truth but are not detected by the edge extraction technique.

Furthermore, to prove the robustness contributed by the segmentation step to the edge extraction process, we have compared our technique with the method in (Bazazian et al., 2015). To this end, we consider an unorganized point cloud which is a Multiple-Tetrahedron, as shown in Figure 10. We propose this shape for two reasons: first, as a synthetic shape, the ground-truth of edge points is available. Second, it comprises 4 different tetrahedrons of different surface sizes, to demonstrate the advantage of the segmentation approach on the different sizes of each segment. Table 1 shows the F1-score: we compare the accuracy of our proposed technique (which is by segmentation) with the technique of (Bazazian et al., 2015) (which is without segmentation). We demonstrate that performing segmentation as a pre-process of the edge extraction yields to higher accuracy. By segmenting the point cloud into different patches, we avoid taking non-local neighbors into account. In addition, by considering an appropriate scale for each segment of a point cloud, we eliminate false positive results which are due to the scale incompatibility. As shown in Table 1, by employing segmentation as a prerequisite of edge extraction process we obtain a higher accuracy. Moreover, in order to examine the robustness of our proposed method to noisier real-world datasets, we show the results for noisy models perturbed with additive 10% Gaussian noise. As it is shown qualitatively in Figure 9 the edges are extracted accurately, although there are noisy points surrounding the Bunny. Thus, the results in this case are almost the same as in the model without noise. Furthermore, to verify the results quantitatively, we have added 10% Gaussian noise to the Multiple-Tetrahedron and computed the accuracy. We have evaluated our proposed method using the F1-score. In Table 2 it is shown that even

when adding noise the accuracy of edge extraction with segmentation is higher than without segmentation.

Table 1: Comparing the accuracy of edge extraction for the Multiple-Tetrahedron figure with segmentation (our proposed algorithm) and without segmentation (the technique of (Bazazian et al., 2015)).

Multiple Tetrahedron	Precision	Recall	F1-Score
With Segmentation	0.786	0.825	0.805
Without Segmentation	0.651	0.842	0.734

Table 2: Evaluating the accuracy of edge extraction for the Multiple-Tetrahedron model perturbed with 10% additive Gaussian noise. Our proposed method refers to the line where calls with segmentation, whereas the method proposed in (Bazazian et al., 2015) is called by without segmentation.

Multiple Tetrahedron with Noise	Precision	Recall	F1-Score
With Segmentation	0.476	0.824	0.603
Without Segmentation	0.405	0.826	0.543

## 5 CONCLUSION

In this paper we have presented a segmentation-based technique for multi-scale edge extraction. We have focused on the challenge of edge detection when foreign surfaces get into local analysis neighborhoods. A global segmentation based on geodesic distance allows segmenting different regions of a point cloud. This allows exploring the best scale for each segment and improves the accuracy of edge extraction.

We have illustrated the importance of determining the best scale as a prerequisite for the edge extraction process. Consequently, we propose a multi-scale technique that finds the appropriate neighborhood size for the analysis, instead of setting a single scale for whole the point cloud. This is the basis for the superior accuracy of the proposed method in the quantitative evaluation results we present.

We have also proven that the segmentation of the point cloud, makes feasible to define the scale of each segment before finding the optimal neighborhood size. This eliminates false positives due either to the incompatibility of neighborhood sizes and segment scales or non-local neighbors. Accordingly, it yields an improved precision in edge detection.

We have quantitatively compared the accuracy of results in the analysis of synthetic objects for which



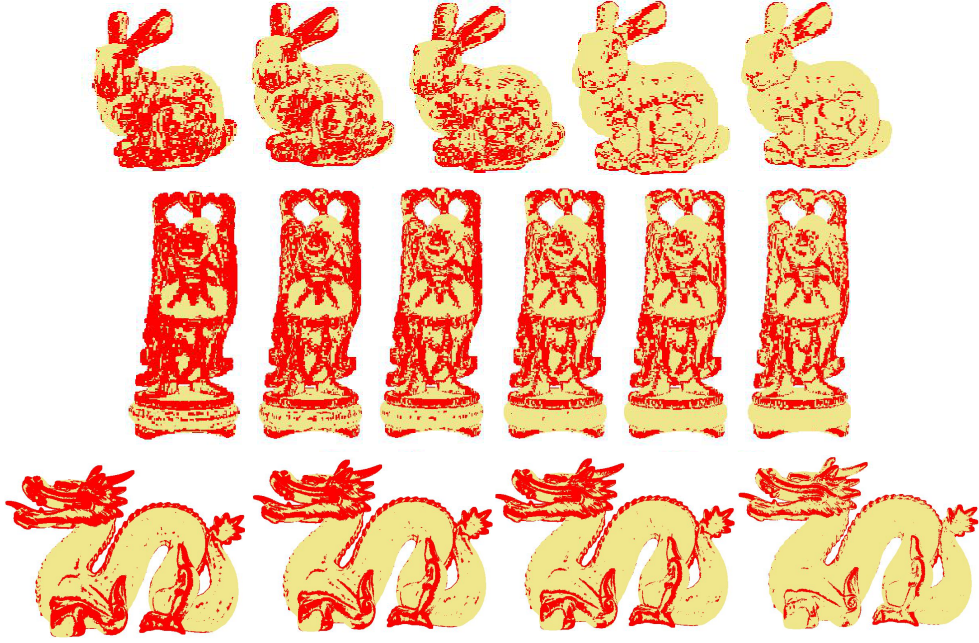


Figure 5: Iteration of Extracting Edges for the point clouds of Bunny, Buddha and Dragon

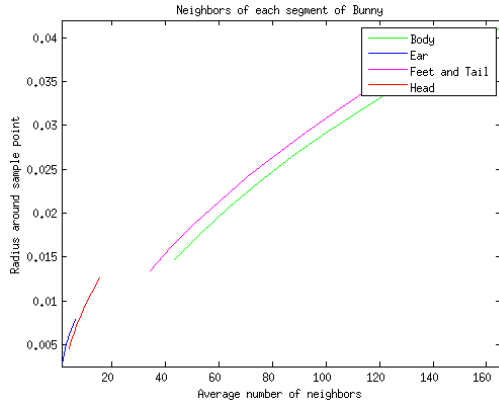


Figure 6: Plot of the Average of the area and number of neighbors for each segment of the Bunny

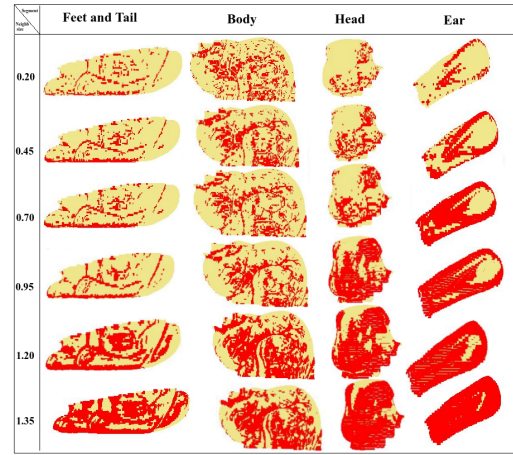


Figure 7: The different segments of Bunny behave differently for edge extraction with progressive neighborhood sizes

we do have ground truth. Furthermore, by adding Gaussian noise to the artificial point clouds, we have demonstrated that the presented approach is more robust in noisier real-world datasets.

Experimental results show quantitatively and qualitatively that our algorithm can deal with different unorganized point cloud surfaces, can complete surface segmentation and then apply multi-scale edge extraction robustly.

Moreover, the proposed algorithm is able to determine independently the neighborhood size of each segment in a point. This makes it possible to perform a precise and semi-automatic edge extraction proce-

dures for point cloud analysis.

The contributions of this paper are in two main aspects: first, we declare that considering a single neighbor size for all the segments of a point cloud is prone to error, particularly for small segments. Therefore, in order to have an accurate edge extraction system, it is essential to assign an appropriate neighbor size for each segment of the point cloud. The second aspect is that we propose a robust semi-automatic edge extraction algorithm based on surface segmentation.

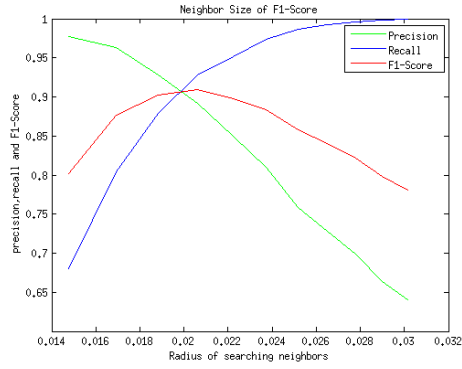


Figure 8: Comparison of the F1-Score for the various neighborhood sizes in order to explore the adaptive scale for a segment of the point cloud corresponding to the intersection of 3 planes

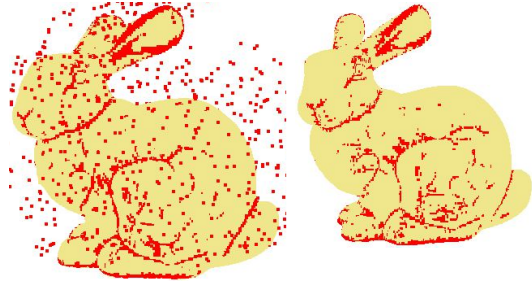


Figure 9: Comparing the Edge Extraction technique of Bunny with Gaussian Noise and without noise

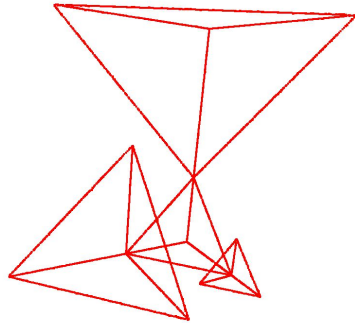


Figure 10: The geometric shape of Multiple-Tetrahedron

As future work, we aim to exploit the algorithm proposed in (Rieck and Leitte, 2014) in order to analyze the skeleton of a surface according to the persistent homology in order to obtain the multi-scale description in a point cloud.

## ACKNOWLEDGMENT

This work has been developed in the framework of projects TEC2013-43935-R and TEC2016-75976-

R, financed by the Spanish Ministerio de Economía y Competitividad and the European Regional Development Fund (ERDF).



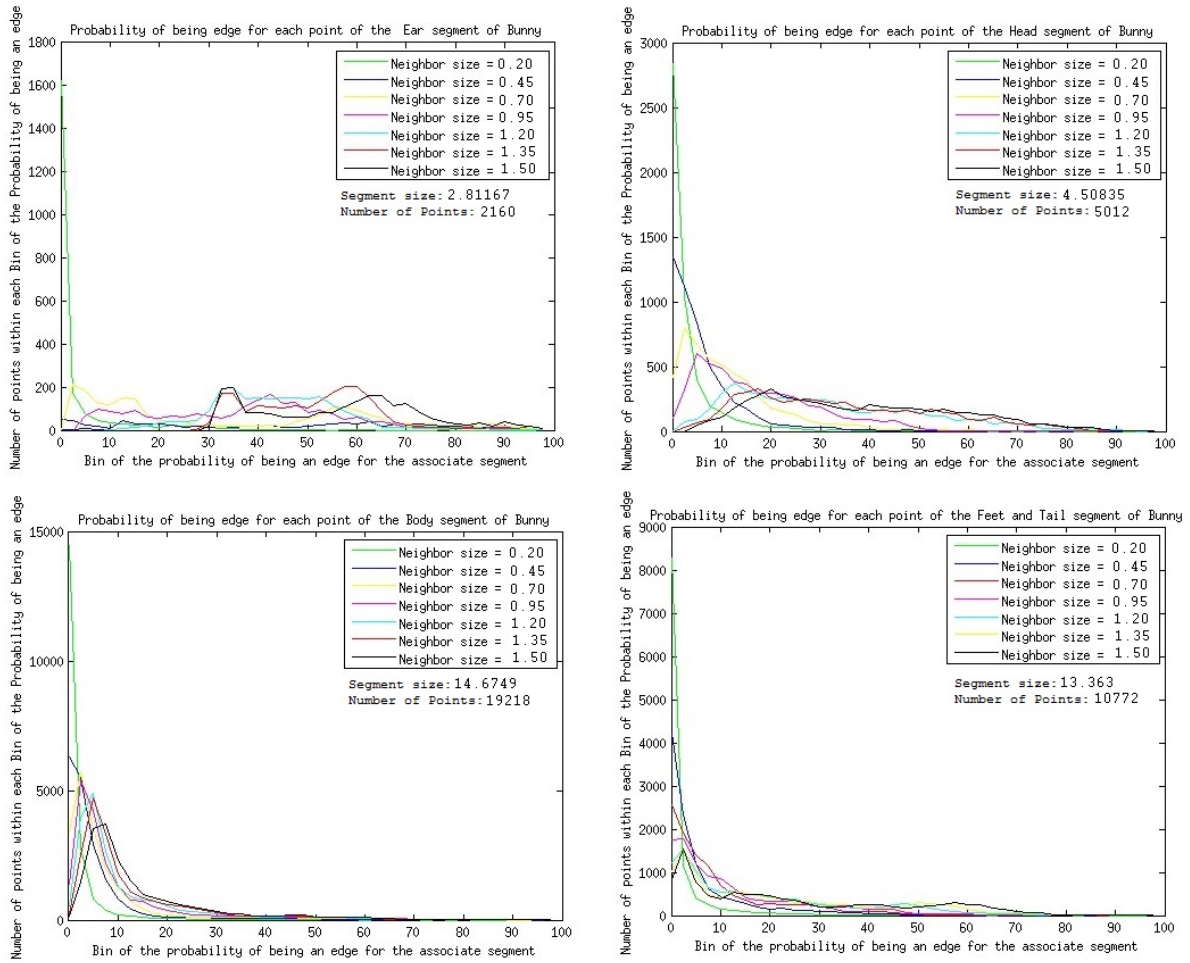


Figure 11: Probability of being an edge according to the various segments of the Bunny over various scales

## REFERENCES

- Bazazian, D., Casas, J., and Ruiz-Hidalgo, J. (2015). Fast and robust edge extraction in unorganized point clouds. *Digital Image Computing: Techniques and Applications (DICTA)*.
- Choi, C., Trevor, A. J. B., and Christensen, H. I. (2013). RGB-D edge detection and edge-based registration. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1568–1575.
- Daniels, J., Ochotta, T., and Ha, L. (2008). Spline-based feature curves from pointsampled geometry. *Vis. Comput.*, 26(6):449–462.
- Demarsin, K., Vanderstraeten, D., Volodine, T., and Roose, D. (2007). Detection of closed sharp edges in point clouds using normal estimation and graph theory. *Computer-Aided Design*, 39(4):276–283.
- Feng, C., Taguchi, Y., and Kamat, V. (2014). Fast plane extraction in organized point clouds using agglomerative hierarchical clustering. *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6218–6225.
- Fleischman, S., Cohenor, D., and Silva, T. (2005). Robust moving least-squares fitting with sharp features. *ACM Trans Graph.*, pages 37–49.
- Gross, M. and Pfister, H. (2007). Point-based graphics (the morgan kaufmann series in computer graphics). *San Francisco, CA, USA: Morgan Kaufmann Publishers*.
- Gumhold, S., Wang, X., and Mcleod, R. (2001). Feature extraction from point clouds. *Proceedings of 10th International Meshing Roundtable*.
- Hackel, T., Wegner, J. D., and Schindler, K. (2016). Contour detection in unstructured 3d point clouds. In *CVPR 2016*, volume 3, Las Vegas, NV, US.
- Ho, H. and Gibbines, D. (2009). Curvature-based approach for multi-scale feature extraction from 3d meshes and unstructured point clouds. *IET Computer Vision*, 3(4):201–212.
- Hoppe, H., Deroose, T., Duchamp, T., McDonald, J., and Stuetzle, W. (1992). Surface reconstruction from unorganized points. In: *ACM Siggraph*, page 7178.

- Ioannou, Y., Taati, B., Harrap, R., and Greenspan, M. (2015). Difference of normals as a multi-scale operator in unorganized point clouds. *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT)*, pages 501–508.
- Kustra, J., Jalba, A., and Telea, A. (2014). Robust segmentation of multiple intersecting manifolds from unoriented noisy point clouds. *Computer Graphics Forum*, 33(1):73–87.
- Lange, C. and K., P. (2005). Anisotropic smoothing of point sets. *Comput Aided Geom Des*, 22(7):680692.
- Lin, Y., Wang, C., Cheng, J., Chen, B., Jia, C., Chen, Z., and Li, J. (2015). Line segment extraction for large scale unorganized point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 102:172–183.
- Liu, Y. and Youlun, X. (2008). Automatic segmentation of unorganized noisy point clouds based on the gaussian map. *Computer-Aided Design*, 40(5):576–594.
- Mitra, N. and Nguyen, A. (2003). Estimating surface normals in noisy point cloud data. *Proceedings of the nineteenth annual symposium on computational geometry*, pages 322–328.
- Oztireli, C., Guennebaud, G., and Gross, M. (2009). Feature preserving point set surfaces based on non-linear kernel regression. *Computer Graphics Forum*, 28(2).
- Park, M., Lee, S., and Lee, K. (2012). Multi-scale tensor voting for feature extraction from unstructured point clouds. *Graphical Models*, 74(4):197–208.
- Pauly, M., Keiser, R., and Gross, M. (2003a). Multi-scale feature extraction on pointsampled surfaces. *Computer Graphics Forum*, 22(3):281–289.
- Pauly, M., Keiser, R., Kobbelt, L., and Gross, M. (2003b). Shape modeling with point-sampled geometry. *ACM Trans Graph*, 22(3):641650.
- Rabbani, T., van Den Heuvel, F., and Vosselmann, G. (2006). Segmentation of point clouds using smoothness constraint. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.*, 36:248–253.
- Reisner-Kollmann, I. and Maierhofer, S. (2012). Segmenting multiple range images with primitive shapes. *19th Int. Conf. on Systems, Signals and Image Processing*.
- Rieck, B. and Leitte, H. (2014). Structural analysis of multivariate point clouds using simplicial chains. *Computer Graphics Forum*, 33(8).
- Sidiropoulos, A. A. and Lakakis, K. N. (2016). Edge Points Detection in Unorganized Point Clouds. *Int. J. Constructive Research in Civil Engineering*, 2(3):8 – 15.
- Son, H. and Kim, C. (2013). Fully automated as-built 3d pipeline segmentation based on curvature computation from laser-scanned data. *Computing in Civil Engineering*, pages 765–772.
- Wang, Y. and Feng, H. (2015). Outlier detection for scanned point clouds using majority voting. *Computer-Aided Design*, 62:31–43.
- Weber, C., Hahmann, S., and Hagen, H. (2010). Methods for feature detection in point clouds. *Visualization of Large and Unstructured Data Sets IRTG Workshop*, pages 90–99.
- Weber, C., Hahmann, S., Hagen, H., and Bonneau, G. (2012). Sharp feature preserving mls surface reconstruction based on local feature line approximations. *Graphical Models*, 74(6):335–345.
- Xu, J., Zhou, M., Wu, Z., Shui, W., and Ali, S. (2015). Robust surface segmentation and edge feature lines extraction from fractured fragments of relics. *Journal of Computational Design and Engineering*, 2(2):79–87.
- Zhang, G., Vela, P., and Brilakis, I. (2013). Detecting, fitting, and classifying surface primitives for infrastructure point cloud data. *ASCE Computing in Civil Engineering*, pages 589–596.